ORIGINAL SCIENTIFIC PAPER

CROATICA CHEMICA ACTA

Croat. Chem. Acta **2016**, *89*(4), 481–492
Published online: March 12, 2017
DOI: 10.5562/cca3027

HKD 1926

Superposing Significant Interaction Rules (SSIR) Method: A simple Procedure for Rapid Ranking of Congeneric Compounds

Emili Besalú,[1,*] Lionello Pogliani,[2] Jesús Vicente de Julián-Ortiz[3]

[1] Institut de Química Computacional i Catàlisi (IQCC) and Departament de Química, Universitat de Girona, C/ Maria Aurelia Capmany, 69, 17003 Girona, Catalonia, Spain

[2] Unidad de Investigación de Diseño de Fármacos y Conectividad Molecular, Departamento de Química Física, Facultad de Farmacia, Universitat de València, Burjassot (València), Spain, and MOLware SL, Valencia, Spain

[3] ProtoQSAR SL, Parc Científic, 46980 Paterna / Departament de Química Física, Facultat de Farmàcia, Universitat de Valencia, Av. V. Andrés Estellés 0, 46100 Burjassot, Valencia, Spain

* Corresponding author's e-mail address: emili.besalu@udg.edu

RECEIVED: October 19, 2016 ⋆ REVISED: November 29, 2016 ⋆ ACCEPTED: December 20, 2016

THIS PAPER IS DEDICATED TO PROFESSOR NENAD TRINAJSTIĆ ON THE OCCASION OF HIS 80TH BIRTHDAY

Abstract: The Superposing Significant Interaction Rules (SSIR) method is revised and implemented. The method is a simple combinatorial procedure, which deals with *in situ* generated rules among a dichotomized congeneric molecular family, selecting the most probabilistically relevant ones. The mere counting of the number of relevant rules attached to new compounds generates a molecular ranking useful for database filtering, refinement and prediction. The algorithm only needs for a symbolic molecular representation and this allows for mining the database in a confidential manner. Third parties will not know the real compounds that are on the way to be worked out. The procedure is tested for a complete series of substituted amino acids. Areas under the receiver operating characteristic (AU-ROC) are always greater than 0.9 for all the following tried protocols: training, leave-one out, balanced leave-two-out and 5-fold cross validations and, finally, a stochastic series of calculations combined with a randomization test.

Keywords: SSIR method, Congener series, Ranking, SAR, Balanced Leave-two-out cross validation (BL2O).

INTRODUCTION

DESPITE that several methods exist to mine data of congener series or combinatorial data sets,[1–5] none has been shown to be as simple as the Superposing Significant Interaction Rules (SSIR) method.[6,7] Despite not being a quantitative method, this simple and systematic procedure is able to rank analogous series. A seminal basic procedure that inspired SSIR was based on a substitutions frequency analysis, leading to good results when dealing with small peptide libraries.[8–10] The algorithm genesis was inspired by the Design of Experiments (DoE) theory,[11] but seeking for a general procedure avoiding to deal with predefined orthogonal designs (*i.e.*, avoiding to synthesize a series of predefined analogues). Instead, the present method allows working with the database 'as it is'.

SSIR procedure is able to work with congeneric series presenting many substitution sites (named factors in DoE terminology), each one able to accommodate an (almost) arbitrary number of substituents (or levels). The procedure systematically quantifies the importance of substituent interactions and extracts relevant information even for unbalanced (arbitrary) libraries. The method is also well suited for the treatment of molecular sets described by fingerprints (*i.e.*, series of categorical or rank descriptors). In this case it is not necessary to deal with a congener series and the method becomes applicable to arbitrary molecular families. Additionally, the process allows dealing with confidential data because the symbolic codification permits the data owner to distribute them in a masked way.

EXPERIMENTAL SECTION: GENERAL FRAMEWORK

Libraries and Sublibraries

Usually, a congener series will be defined after the presence of a common core structure along a molecular family. Other approximation relies on the process of molecular alignment and ulterior definition of equivalent substitution sites. In both cases the molecular family can be represented schematically as a set of n substitution sites where, in turn, each element i is constituted by a set of m_i possible substituents (A, B, C, ...). Within the chemical framework, some of the substituents can be repeated in different sites, but, in general, their chemical influence is distinct. As the molecular positioning of substituents is relevant, one is free to use the same set of symbols (A, B, C,...) to represent each series of substituents provided that the anchorage points (1, 2, ..., n) make the sets diverse in essence. The list of substituents per site expands the combinatorial universe of compounds, the number of them being

$$M = \prod_{i=1}^{n} m_i \tag{1}$$

This corresponds to the cardinal of the following Cartesian product:

$$\{A,B,C,..\}_{m_1 \text{ elements}} \times \{A,B,C,..\}_{m_2 \text{ elements}} \times ... \times \{A,B,C,..\}_{m_n \text{ elements}}$$

The set of n-tuples as *AAA...A* or *EBD...Z* constitutes the whole database or the entire molecular universe representation.

This paper deals with structure-activity relationship (SAR) ranking rules along sublibraries, which are subsets of the complete set of M elements. In practice, these sublibraries were generated according to a particular history and are usually disperse and inhomogeneous. That is the reason why the available set of molecules or subsets of them normally does not correspond to a homogeneous or systematic (orthogonal) sublibrary as those required either by DoE or by D-Optimal designs.[11]

The Hypothesis

SSIR method is based on the assumptions that the interactions among substituents (factors) play an important role and that the activity is not due to a simple substituent but to the cooperative effect of several residues. Prior to the method application, a binary encoding must be set into the database. Sometimes the partition will be natural (*i.e.*, the variable or property of interest is by itself binary), and in other cases (e.g. for categorical or continuous variables) some congeners are arbitrarily labeled as being of interest (e.g. be active, be a drug, give high signal, etc.), while others as molecules of not interest. This partition will depend on a fixed threshold value set by the researcher along the studied molecular property.

SSIR method consists into loop over combinations of k = 1, 2, 3, ... sites, and for each combination loop over all the possible variations of residues. Each sequence of k residues conforms a *variable* or *rule* of order k that is attached to a library subset. It is said that the rule *condenses* this subset of molecules (*i.e.*, the congeners fulfill the rule) and some of them will be of interest according to the previously fixed database partition. Then, each rule is evaluated from a probabilistic point of view, according to the number of condensed analogues of interest. Significant rules are kept and a vote is attached to each one. Normally, many probabilistically significant rules are found and the superposition or combination of them results in a synergic positive effect giving a clue to 'point' towards new interesting derivations. In other words, each training, test or validation compound will have a score coming from the votes of the rules that condenses it. It is expected that the higher the score the higher the probability for the molecule to be useful.

Generation of Rules: Tracking Interactions

For illustrative purposes, a toy library is here considered. A full database or molecular universe can be obtained by combining four residues in n=4 sites. The full set of molecules is given by the Cartesian product R=R_1×R_2×R_3×R_4 where the site substituents are represented by the sets R_1={*A*,*B*,*C*}, R_2={*A*,*B*,*C*,*D*}, R_3={*A*,*B*,*C*,*D*} and R_4={*A*,*B*,*C*,*D*,*E*}. Their cardinals being m_1 = 3, m_2 = m_3 = 4 and m_4 = 5. Hence, the complete library R has, according to equation (1), M=3×4×4×5=240 congeners: R={*AAAA*, *AAAB*, *AAAC* ,..., *CDDD*, *CDDE*}. Note that the combinatorial database codification is made up from arbitrary symbols.

It is convenient to define the whole available set of residues attached to a particular site i. This set is denoted by the wildcard X or as the set $\{X\}_i$. Under this notation, the full database can be denoted by R = $\{X\}_1$×$\{X\}_2$×$\{X\}_3$×$\{X\}_4$ = {*XXXX*} = *XXXX*. The last symbol is the rule defining the set of molecules having any of the available residues at the corresponding positions, i. e., the entire library.

As said, partial subsets of molecules from the database are specified (collected or condensed) by a rule of a certain order k. For instance, a possible set of combinations of sites are those involving the second and the fourth, for which the list of combinations are denoted by the Cartesian product $\{X\}_1$×R_2×$\{X\}_3$×R_4. This pattern gives a total of m_2×m_4=4×5=20 rules of order 2. These rules identify the variations of residues at positions 2 and 4

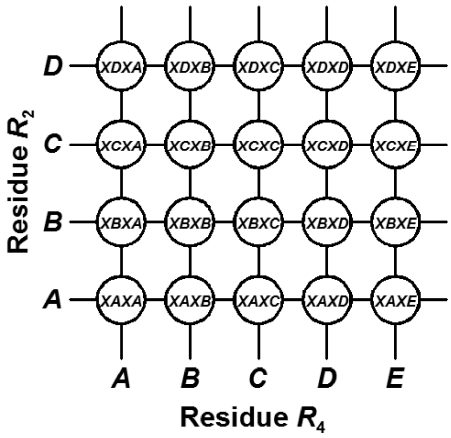

Figure 1. The example of full arrangement of second order rules, each one condensing 12 molecules.

disregarding of the substitutions at positions 1 and 3 and are *XAXA*, *XAXB*, and so on up to *XDXE*.

Figure 1 shows a grid arranging all these 20 rules, each one collecting or condensing a maximum of $m_1 \times m_3 =$ 12 compounds. For instance, the rule *XAXA* is understood as the set of compounds having the *simultaneous* substitutions $A \in R_2$ and $A \in R_4$ at positions 2 and 4, respectively:

$$XAXA = \{AAAA, AABA, AACA, AADA, BAAA, BABA, BACA, BADA, CAAA, CABA, CACA, CADA\}$$

In the combinatorial chemistry field[12] there are several methods to systematically generate libraries in the laboratory that are related to the synthesis of mixtures of the compounds found in each of the rules appearing in Figure 1. Within the experimental context, the notation *X* at a certain rule position stands for a mixture of all compounds presenting all the possible substituents at that position. For instance, for a synthetic combinatorial chemist the variable *XAXA* can mean a mixture of the 12 compounds listed above. In the laboratory it is very common to obtain these mixtures in a progressive robotized framework, which in turn can lead to the knowledge of the mean activity of the mixture. At the end of programmed sequences, when arriving at a rule "without *X*s" standing for a single compound, the activity of this particular compound can also be known.

In general, a library arising from a scaffold presenting n substitution sites, $R=R_1 \times ... \times R_n$, generates a total of $\binom{n}{k}$ combinations of k sites. Each combination generates a fixed number of variables of order k. If the sites are able to accommodate m_1, m_2, ..., m_n moieties, for the case of rules of order 2 there are $\binom{n}{2}$ combinations of two sites and

- Combination 1 expands sites 1 and 2, and generates $m_1 m_2$ rules, each one condensing a maximum of $m_3 \cdot m_4 \cdots m_n$ compounds.
- Combination 2 expands sites 1 and 3, and generates $m_1 m_3$ rules, each one condensing a maximum of $m_2 \cdot m_4 \cdots m_n$ compounds.
- ...
- Combination $n(n-1)/2$ expands sites $n-1$ and n, and generates $m_{n-1} m_n$ rules, each one condensing a maximum of $m_1 \cdot m_2 \cdots m_{n-2}$ compounds.

The mathematical expression giving the total number of rules, V, provided that there are not redundant symmetry issues (which of course it may occur in chemistry), is

$$V = \sum_{k=1}^{n} \left\{ \sum_{i_1=1}^{n-k+1} \sum_{i_2=i_1+1}^{n-k+2} \cdots \sum_{i_k=i_{k-1}+1}^{n} \left(\prod_{j=1}^{k} m_{i_j} \right) \right\} \quad (2)$$

In Eq. (2) the outer leftmost summation defines the order of the interaction of the rule. In practical applications usually only low rule orders will be explored (as it is found in the DoE field, interactions of low order uses to be the main responsible of the molecular responses). The series of the inner k summation symbols generates the combinations of k elements taken from the pool of n (*i.e.*, the selection of involved sites in each variable of order k). Finally, the rightmost product involving k terms counts how many variables are generated from the previously selected k substitution sites. This corresponds to permutations with repetition. Given the values of i_1, i_2,...,i_k, (*i.e.*, the identification of the sites being combined) the number of generated rules is

$$v_k = v(i_1, i_2, ..., i_k) = \prod_{j=1}^{k} m_{i_j} \quad (3)$$

The maximum number of compounds being condensed by each one of these rules is[6]

$$c_k = c(i_1, i_2, ..., i_k) = \prod_{j \neq i_1, i_2, ..., i_k}^{n} m_j = \left(\prod_{j=1}^{n} m_j \right) \Big/ \left(\prod_{j=1}^{k} m_{i_j} \right) = \frac{M}{v_k} \quad (4)$$

Notice that, for each order k, the full generation of rules and the total count of compounds being condensed by them gives the total number of potential molecules in the library, *i.e.*, $M = c_k v_k$. In practical applications, the total number of molecules belonging to the library or being condensed by a rule may not be available or yet synthesized. That is the reason why we are using a terminology in terms of maximum number of compounds.

Expressions (2)–(4) are attached to variables for which only the 'presence of a residue' concept is being considered. The number of variables increases if rules involving the concept of 'non presence of a residue' are taken into account. The *negation* of a certain residue will be denoted by a bar. In our example, the set of molecules *not* having residue *A* at position 1 is defined by the rule $\overline{A}XXX$. This stands for the set {*BXXX*,*CXXX*} or, equivalently, for a difference of two sets: {*XXXX*}\{*AXXX*}. The combination of two complementary (non exclusive) rules can lead to other specific sets. Following with our example in terms of set notation, the following rule (5) stands for the sublibrary of molecules not having residue *A* at the first site and *simultaneously* not having the residue *B* at the second anchorage point.

$$\overline{A}\overline{B}XX = \overline{A}XXX \cap X\overline{B}XX \tag{5}$$

That is the same as the set,

$$\{B,C\}\times\{A,C,D\}\times\{A,B,C,D\}\times\{A,B,C,D,E\} = (R_1 \setminus \{A\})\times(R_2 \setminus \{B\})\times R_3 \times R_4$$

or, equivalently, the Cartesian product

$$\{B,C\}\times\{A,C,D\}\times\{X\}\times\{X\} = \{BAXX,BCXX,BDXX,CAXX,CCXX,CDXX\} = BAXX \cup BCXX \cup BDXX \cup CAXX \cup CCXX \cup CDXX.$$

In reference 6 more details concerning the algebra of rules notation are given. Despite the algebra opens the possibility to define many combinations of rules, in this work all the rules will only involve juxtaposed positive (*A*, *B*, ...) and negation ($\overline{A}$, $\overline{B}$, ...) individual terms. Hence, here are only considered rules like *ABXX* = *AXXX* ∩ *XBXX* or rules involving individual negation operators, as in,

$$\overline{A}\,\overline{B}C\,\overline{D} = \overline{A}XXX \cap X\overline{B}XX \cap XXCX \cap XXX\overline{D}$$

Under these restrictions, the systematic computational generation of rules is performed nesting three combinatorial entities (see the SSIR basic training algorithm below). The first one explicitly generates combinations among *k* sites in order to set up the rule order, the second one generates the permutations with repetition among residues attached to the previously selected sites. Finally, a third one may eventually generate 2^{k-L} binary numbers, constituting a flag for the codification of all the possible combinations of negations and non-negations. All these algorithms are well known in the field of discrete mathematics.[13–16]

Regarding the generation of binary numbers, it must be mentioned that the negation flags are not to be systematically applied in all the sites (hence the superindex -*L* above).[6,7] The sites presenting only two possible substitutions (*i.e.* binary sites) are to be avoided from this choice because one level is the natural negation of the other one. In this case the negations are generated implicitly. Particularly, for rules of order 1 involving a binary site, only one of the levels has to be considered. This is so because this rule, if being interesting form the probabilistic point of view, will bear a positive or a negative vote (see below) and this information is enough and complete because the other (complementary) rule will automatically bear a negative or a positive vote, respectively, being the information wholly redundant if both rules are kept.

Probabilistic Significance Attached to a Variable

SSIR method does not take into account every generated or definable rule. Only those attached to a significant probability are able to enter into the final SAR model. The probability of a rule comes from and is influenced by the molecular library partition described above. The significance of a rule is defined from the hypergeometric formula.[17–19] Let us suppose that the library is composed by $a \le M$ known molecules, and *b* of them are of interest. Then, if a rule condenses *c* of those known compounds, *d* of them being also of interest, one can ask for the probability of this event (see Figure 2a for a simple representation of the subsets involved). This probability can be evaluated if the independence of events is assumed (this is particularly true for true random databases, but not strictly for focused ones) and it is given by the following conditional probability:

$$P(d,c|b,a) = \frac{\binom{b}{d}\binom{a-b}{c-d}}{\binom{a}{c}} \tag{6}$$

with $d \le c \le a \le M$ and $d \le b \le a \le M$

where *d* ranges from max(0,*c*+*b*−*a*) up to min(*b*,*c*). The significance level, or *p*-value, attached to each rule is obtained from the cumulated probabilities that the rule condenses *d* or more (*d*+) structures of interest:

$$p(d+,c|b,a) = p(d:\min(b,c),c|b,a) = \sum_{i=d}^{\min(b,c)} P(i,c|b,a) = 1 - \sum_{i=\max(0,c+b-a)}^{d-1} P(i,c|b,a) \tag{7}$$

Berkopec's algorithm[20] is very useful to compute the result of equation (7) in a fast way and for valid arbitrary values of *a*, *b*, *c* and *d*.

Following our toy example, despite the whole library is virtually formed by *M* = 240 structures, we will assume now that only *a*=200 are at our disposal with known activity, and that in this subset *b* = 80 structures are declared as being of interest. Let us also assume that the

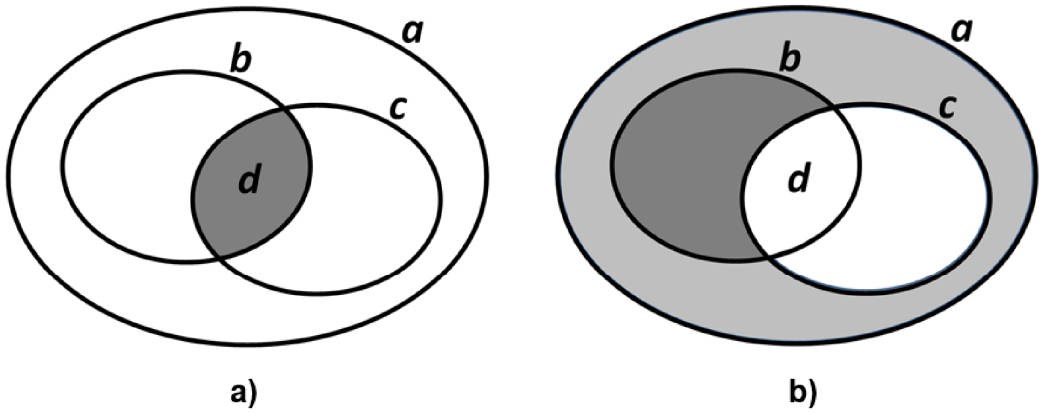

Figure 2. Venn diagrams representing the selection of sets entering into the hypergeometric probabilistic term definitions. a) Direct selection of *d* active compounds (grey zone) when selecting *c* items when it is known that the global set of *a* items encompasses *b* actives. b) Equivalent complementary selection of *b*–*d* active compounds (dark grey zone). The probability of selection of these zones is equivalent, as explained in text.

aforementioned variable *XAXA* condenses *c* = 8 known molecules (simulating that the four remaining ones out of twelve are not yet synthesized) being *d* = 7 of them active. We see that the proportion *b* / *a* = 80 / 200 is less than *d* / *c* = 7 / 8. This tells us that the variable condenses molecules of interest at a superior ratio than the ratio found in the entire known subset. This enrichment is desirable in order to infer about the performance of the rule *outside* the known sublibrary. The above probabilistic calculation will say about the importance of the rule. Equation (6) tells that, assuming uniformity of events, the probability to collect 7 active molecules is $P(7,8|80,200) = 0.0069$. According to equation (6), the probability to collect 7 or more active molecules is $p(7+,8|80,200) = p(7:8,8|80,200) = P(7,8|80,200) + P(8,8|80,200) = 0.0074$. Thus, variable *XAXA* is a good candidate to enter into our model because it condenses structures of interest at a significant ratio attached to a probability of only 0.74 % to get equal or better results when selecting a random subset of 8 structures in our database. The ultimate goal is to guess for new unknown compounds being active. It is assumed that the accumulation of several significant rules provides a scoring method in order to increase the chances to point to *new* active derivatives outside the known portion of the database.

Positive and Negative Votes

In our example above, the rule *XAXA* should be assigned a positive vote or score. But there are variables able to enter into the model which can be attached to negative votes. This will be the case for a rule, say *BCXX*, that condenses *c* = 10 known molecules and only one of them is active (*d* = 1). The probability to collect one or more active molecules when randomly selecting 10 compounds in our database is given by the expression: $p(1+,10|80,200) = p(1:10,10|80,200) = 0.9948$. This large value indicates that the *BCXX* rule should apparently be avoided because it mainly concerns molecules of no interest. Fortunately, this is still a helpful information. Notice that the sum of the entire range of probabilities *P* adds up the unit:

$$p(\max(0,c+b-a):\min(c,b),c|b,a) = 1$$

This is because it covers the full range of mutually exclusive possibilities. Hence, it is immediate to see that the expression

$$p(d-,c|b,a) = 1 - p(d+,c|b,a) + P(d,c|b,a)$$

indicates that the probability to select *d* or less (*d*−) molecules of interest can be a small number provided that $p(d+,c|b,a)$ is big and $P(d,c|b,a)$ is small enough. In our example, as $P(1,10|80,200) = 0.0373$, it is found $p(1-,10|80,200) = 0.0424$. For many purposes this will be a significant result. As a consequence, if desired, the variable *BCXX* could be also included in the model but attached to a *negative* punishing vote. It is expected that new molecules having (only) this variable pattern will probably be of no interest. Of course, this is not a forced event, but it is expected that the accumulation of several rules is acting in a synergic way helping to determine some of the characteristics that a (new) analogue of interest must have.

The general relationship $\binom{n}{m} = \binom{n}{n-m}$ states that selecting *m* objects from a set of *n* identical ones is the complementary action of selecting *n*−*m* from *n*. A similar rule relates the $P(d,c|b,a)$ values: in a set of *a* items being *b* of them of interest, selecting *c* among *a* automatically forces to make a complementary selection of *a*−*c* among the same set of *a*. At the same time, if along the selection

Table 1. Series of the possible number of active molecules that can be found along the set of known structures (a congeners being b of them of interest). See text for the symbols. The numbers depicted at the bottom are those corresponding to the example explained in the main text (a = 200, b = 80) when c = 10 and d = 1

Increasing number of active items in original set →									
max(0,c+b−a)	max(0,c+b−a)+1	max(0,c+b−a)+2	...	d-1	d	d+1	...	min(b,c)-1	min(b,c)
min(b,a-c)	min(b,a-c)-1	min(b,a-c)-2	...	b-d+1	b-d	b-d-1	...	max(0,b-c)+1	max(0,b-c)
Decreasing number of active items in complementary set →									
Increasing number of active items in original set (particular toy example case in the main text) →									
0	1	2	...	4	5	6	...	9	10
80	79	78	...	76	75	74	...	71	70
Decreasing number of active items in complementary set (particular toy example case in the main text) →									

of the first c items d of them are of interest, automatically there are found b−d of interest along the subset of a−c. So,

$$P(d,c|b,a) = P(b-d,a-c|b,a). \quad (8)$$

In this context, the index d can range from max(0,c+b−a) increasing up to min(b,c), as indicated by the range of index i in equation (7). Figure 2a schematically shows by means of a Venn diagram what the left part of equation (8) stands for: in a set of a items where b of them are of interest, when a rule selects (condenses) c items it is at the same time selecting d active terms from the set of b (grey zone). This selection procedure is totally equivalent to the interpretation showed in Figure 2b: the selection of the c terms causes the simultaneous and complementary selection of a–c items (grey zones). Of these items, b–d are of interest (dark grey zone).

In the complementary space the index of active items is swept in the set of a–c items and ranges from min(b,a−c) decreasing up to max(0,b−c) along a one-to-one correspondence. The series of equivalent probability partners (*i.e.* complementary events) can be read by columns in Table 1, where each pair of terms generate the same P probability value, as denoted by equation (8). Note that the number of active items in each pair of complementary partners must add up to b. Hence,

$$b = \max(0,c+b-a) + \min(b,a-c) = \min(b,c) + \max(0,b-c).$$

Generally, when doing the p-value computation of a rule, for which the range of 'number of items of interest' is [d,min(b,c)] (see Eq. 7), the range of items of interest available for the complementary rule is [max(0,b−c),b−d] (see Table 1). For this case, both p-values are coincident:

$$p(d+,c|b,a)=p([b-d]-,a-c|b,a).$$

Conversely, if the range of the number of items of interest is [max(0,c+b−a),d], the range of items of interest swept by the complementary rule is [b−d, min(b,a−c)] and

$$p(d-,c|b,a)=p([b-d]+,a-c|b,a). \quad (9)$$

As said, along the above Table 1 series the respective sums of terms in a full row adds up a probability of 1. Due to the one-to-one correspondence among complementary partners, the addition up to the unity can also be written involving the appropriate terms of both series, for instance,

$$1 = p(\max(0,c+b-a):d-1,c|b,a) + p(\max(0,b-c):b-d,a-c|b,a)$$

or

$$1 = p(b-d:\min(b,a-c),a-c|b,a) + p(d+1:\min(b,c),c|b,a)$$

Following with the example of the fictitious rule *BCXX*, it should be noted that, due to Eq. (9), the significant term p(1−,10|80,200) is equal to p(79+,190|80,200). This last p-value is the one attached to the *complementary* or *negation* of *BCXX* rule: its contrary event is the full negation $\overline{BCXX} = \bar{B}XXX \cup X\bar{C}XX$, *i.e.*, the rule which avoids the *simultaneous* combination of substituents *B* and *C* at positions 2 and 4, respectively. This means that the sentence above related to the negative votes that should be addressed to *BCXX* rule could also be reformulated in terms to give positive votes to the complementary rule $\overline{BCXX}$. In this case, the inclusion of the rule $\overline{BCXX}$ in the model with a positive vote is equivalent to assign positive votes to a set of 11 variables of order 2:

$$\overline{BCXX} = \{\, AAXX, ABXX, ACXX, ADXX, BAXX, BBXX, \quad , BDXX, CAXX, CBXX, CCXX, CDXX \,\}$$

For practical purposes, instead of assigning positive votes to several variables conforming a full negation rule, a negative vote is assigned to a single rule, the original *BCXX* variable.

SAR Models and Consensus Votes

The generation of the SAR model involves the assignation of positive or negative votes to each rule declared as significant. Afterwards, each molecular structure will cumulate the votes of all the significant rules that are

condensing it. The ultimate goal is to apply this voting procedure also over new compounds not present in the sublibrary (either with unknown activity or yet not synthesized) and it is expected to rank them properly.

The following algorithm is implemented in the program SSIR[21] and it conforms the basic module of training (getting a model) and model application to an external set.

Algorithm: Basic SSIR procedure for ranking a_{ext} items from the rules trained with a_{trn} items.

1. Input the molecular training data information:
   - 1.1. Read the molecular structure symbolic codifications: a_{trn} congeners entered.
   - 1.2. Set the number of anchorage points per congener: n.
   - 1.3. Set the number of substituents per ancorage point: m_1, m_2,..., m_n.
   - 1.4. Read the molecular data property values and dichotomize it: a total of b_{trn} training molecules are declared of being of interest.
2. Set the range of rule orders to be explored: [k_i , k_f] where $1 \le k_i \le k_f \le n$.
3. Set the threshold p-value per rule order: $p_t(k)$, $k = k_i$, ... , k_f.
4. Generate rules and keep the probabilistically relevant ones:
   - 4.1. Number of rules kept: N_r=0
   - 4.2. Loop for $k = k_i$, k_f. Loop over rule orders: summation over k in equation (2).
     - Loop: For each rule order generate the $C(n,k)$ combinations of sites. This corresponds to k nested loops, i. e., the summations over i_1, i_2, ..., i_k in equation (2).
       - Loop: Generate the variations among the m_{i1}, m_{i2}, ..., m_{ik} elements of each substitution site (rightmost part of equation (2)). This corresponds to k nested loops.
         - Loop: If needed, generate the negation terms. This corresponds to a maximum of k nested binary loops.
         - Count how many training congeners are condensed by the rule: c.
         - Count how many of these condensed congeners are of interest: d.
         - if $p(d+,c|b_{trn},a_{trn}) \le p_t(k)$ then put rule in list of kept rules:
           - N_r=N_r+1: One more kept rule.
           - $V_r(N_r)$=+1: Rule number N_r has a positive vote.
         - else If negative votes are being considered then
           - if $p(d–,c|b_{trn},a_{trn}) \le p_t(k)$ then put rule in list of kept rules:
             - N_r=N_r+1: One more kept rule.
             - $V_r(N_r)$=–1:Rule number N_r has a negative vote.
           - end if
         - end if
       - End Loop of negation terms.
     - End Loop of variations.
   - End Loop of combinations.
   - End Loop 4.2. over rule orders. N_r rules kept as being relevant with positive or negative votes.
5. Loop for L = 1 , a_{ext} : loop over external congeners and collect votes from rules.
   - $v(L)$=0. Number of votes per congener number L.
   - Loop for r = 1 , N_r : loop over accepted rules
     - If rule number r condenses external congener number L then
       - $v(L) = v(L) + V_r(r)$
     - end if
   - End Loop over rules
   - End Loop 5 over external congeners
6. Molecular external set is ranked. Sort congeners according to votes: $v(1) \ge v(2) \ge ... \ge v(a_{ext})$.

The program needs to read the training set (step 1) and the main calculation parameters (range of rules to explore and the respective threshold p-values) in steps 2 and 3. Then, the code generates all the rules definable from the training set (by means of the loops included in step 4). Optionally, negation terms can be included if desired. For each generated rule, the code counts how many training structures condenses (c) and how many of these are of interest ($d \le c$). This information allows the computation of the cumulated hypergeometric probabilities $p(d+,c|b_{trn},a_{trn})$ or, eventually, $p(d–,c|b_{trn},a_{trn})$. Comparing these values against the corresponding p-value threshold allows to assign a positive or a negative vote, if any, to the rule (variables V_r(:) in the algorithm). After the step 4 is completed, a total of N_r rules are kept with the corresponding positive or negative vote. In step 5 a loop starts over the external congeners (this can be also done along the training items, but our philosophy focuses on the application over external items of trained models, as it will be seen en next section). Each external molecule is faced against each kept rule in order to see if the congener is condensed by it or not. If condensation occurs, the compound cumulates the rule vote. At the end of step 5 every external compound will bear a certain number of cumulated votes coming from the previously selected rules during training. Now the external molecular set can be ranked according to these vote scores. An evaluation of the method efficiency can be done if the molecular property values are known for the external congeners (this is the usual case for cross-validation tests, as they will be presented below).

Steps 1–4 of the above algorithm not only conforms the training process, but also a variable (rule) selection procedure. Note that this selection process is (must be) totally independent of the external molecular set. It is understood that training and external sets are mutually exclusive. Of course, the external set has to belong to the same global library, as the selected (trained) rules must be applicable to it. In other words, the available substitution sites and the residues found in the external set are to be also present in the training one.

Cross-validation

The program SSIR implements several cross-validation (CV)[22–24] procedures. Our code is always based on the Internal Test Sets (ITS) paradigm.[17,25–28] This means that every cross-validated training to be done over a subset must start from scratch and the variable selection must be totally independent of the other cross-validation loops or subsequent prediction over a test or validation set. This is accomplished by the above algorithm because the training and the model application are done in distinct molecular and mutually exclusive sets. This notion is nowadays promoted in several places of the literature.[23,24] As seen, in SSIR context variable selection means rule selection. Hence, at every training/test simulation cycle the rules entering into the model are selected from scratch. This is done this way in order to simulate a real-world feature: only after the training is finished and a model is set up, the structures whose property has to be predicted are revealed.

In the above algorithm, steps 1–4 constitute the rule generation from scratch. Steps 5 and 6 serve to apply the SAR model over an external molecular set. This algorithmic structure is useful to do CV calculations following the ITS paradigm: It is only necessary to do a dynamic partition in blocks of the original database, and for each left-out block do the training (*i.e.* select rules and set their votes) with the sole information of the remaining blocks using steps 1–4. Then, each left-out block receives (or cumulates) the corresponding predictions by means of application of steps 5 and 6. In order to speed the process, during the CV cycles it is not necessary to generate all the rules again. It is possible to generate and keep all the rules from the beginning. But it is compulsive to recalculate the rule votes at each cycle. In previous works[6,7] it has been described how SSIR procedure allows the implementation of leave-many-out procedures in a faster way than the explicit leave-and-put cycles with replacement accompanied by the subsequent explicit training and prediction. This is so because the model is obtained by a mere addition of integer votes.

The CV variants we are considering here are of two types: exhaustive systematic and stochastic. The systematic procedures are the Leave-one-Out (L1O) and the Balanced Leave-two-Out (BL2O) ones, being both exhaustive processes. The first one selects one congener at a time and builds a model with the remaining ones. Then, the model serves to assign a certain number of votes to the left-out molecule (i. e., the loop in step 5 of the above algorithm only involves a single compound). Then, the left-out item is replaced and the procedure starts again but selecting the next item to be left apart. The BL2O process is a similar process but two congeners are left-out at the same time (two items involved in the loop of step 5), as it will be explained below. The stochastic procedure chosen is the 5-fold CV, for which the molecular set is randomly partitioned into 5 blocks of equal size. Each block will act as test set while the four remaining ones serve as training set. Despite the 5-fold procedure is also exhaustive once the 5 training-test loops are done, the results depend on the particular molecules that entered in each block. Hence, the whole 5-fold process can be repeated obtaining distinct results. As it will be see below, this feature has been taken into account in order to estimate the variability and stability of the final results.

RESULTS AND DISCUSSION: APPLICATION EXAMPLE

Previous application examples of SSIR method are shown in references 6 and 7. There, molecular non-peptidic congeneric families are studied. In both referred cases molecules present various substitution points and, at the same time, each substitution site is able to allocate several residues. Here an example is presented in order to show how SSIR can be applied in a systematic way for a set of substituted peptides. The example consists on the ranking of a full series of $M = a = 2^9 = 512$ peptides presenting activity against NK1 receptors.[29,30] The activities where codified in percentage (values ranging from 0 up to 100) in such a way that the higher the value, the higher the activity. This complete set was previously studied by means of Formal Inference-based Recursive Modeling (FIRM) by Young and Hawkins[31] and with classical Design of Experiments by Barroso and Besalú[18] arriving to similar conclusions. The peptides structure is

H-**[Arg]**-**[Pro-1]**-**[Lys]**-**[Pro-2]**-**[Gln-1]**-**[Gln-2]**-**[Phe-1]**-**[Phe-2]**-[Gly]-**[Leu]**-[Met]-NH_2

Where in the 9 marked bold positions the L-enantiomers were systematically replaced by the D-enantiomers of the same aminoacid, obtaining all the combinations. Previously cited works show that the most important single point substitution is the terminal [Leu], followed in importance by the single substitutions in [Phe-2], [Phe-1], and [Gln-2],

Table 2. Relevant rules of order 1 for the set of neuropeptides. Only the relevant substitution sites are listed in the table header. L stand for L-enantiomers. Dots stand for wildcard (*X*) rule elements. See text for more details

Rule	*p*-value	Vote	[Gln-2]	[Phe-1]	[Phe-2]	[Leu]
1	$1.7 \cdot 10^{-21}$	+	.	.	.	L
2	$6.8 \cdot 10^{-16}$	+	.	.	L	.
3	$3.7 \cdot 10^{-13}$	+	.	L	.	.
4	$6.7 \cdot 10^{-12}$	+	L	.	.	.

in this order. In Barroso and Besalú's work the substitutions in [Gln-1] and [Pro-2] appear to be a secondary interaction. This agrees with the results of Young and Hawkins, as they affirm that the single substitution in these places is not enough to decide which position is more important. Young and Hawkins argue that second order interactions should play an important role at this level of description. As it is shown below, this feature is related to the fact that SSIR rules of order 2 are the most relevant ones found in this study.

In this example, the application of SSIR methodology is straightforward. It is enough to define which are the active molecules of interest to delineate the database binary encoding. This was arbitrarily set as those presenting a 30 % of activity or more (152 structures). At the same time it was set a cutoff *p*-value of 10^{-10}. Hence, the declared relevant rules will be those having a *p*-value equal or less than 10^{-10}. The total number of rules of order 1 that can be generated is 9 (one per site, *DXXXXXXXX*, *XDXXXXXXX*, and so on). Recall that for this set described by binary substitutions it is not necessary to generate (redundant) negations as, for instance, the rule *LXXXXXXXX* gives the same information as the *DXXXXXXXX* one. Only 4 rules of order 1 are significant, i. e., having *p*-values less than 10^{-10} (see Table 2). A positive vote was assigned to these four rules that favors the presence of the L-aminoacids at any of the last four substituted positions. The ordering for preference of the rules, according the respective *p*-values, are the ones attached to the sites [Leu], [Phe-2], [Phe-1] and [Gln-2]. It is worth noting that a simple calculation that only took a few seconds leads to the same main conclusions referred in the above articles, focusing the attention on the relevant sites. It has to be told that performing the SSIR calculation requires minimal preparatory actions, as the residues codification is symbolic and arbitrary.

Both, the training fit and the Leave-one-out CV (L1O-CV) procedures gave the same results: area under the Receiver Operating Characteristic (AU-ROC) curve equals 0.927 (Accuracy = 92.2 %, Sensitivity = 89.5 %, Specificity = 93.3 %, Precision = 85.0 %, Matthew's CC = 81.6 %) The hit rate at 5 % was 96.9 % with an enrichment factor of 3.3, the maximum being 3.4. This hit rate corresponds to select the first 32 ranked molecules and found that 31 of them are of interest. The ROC curve depicted in Figure 3 shows how the method can be a good preliminary tool able to filter a database. The graph becomes segmented because only five kinds of cumulated votes were generated by molecule (0, 1, 2, 3 or 4).

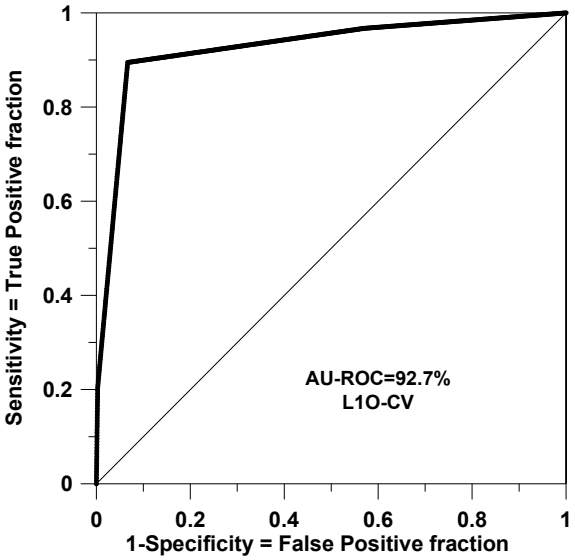

Figure 3. AU-ROC curve and AU-ROC area for the leave-one-out training of the set of neuropeptides considering rules of order 1. See text for details.

A total of $\binom{9}{2}4 = 144$ rules of order 2 can be defined in this set and 15 of them have *p*-values less than 10^{-10}. For training it is obtained AU-ROC = 0.947 (Accuracy = 93.0 %, Sensitivity = 88.2 %, Specificity = 95.0 %, Precision = 88.2 %, Matthew's CC = 83.2 %, Hit rate at 5 % = 96.9 % with an enrichment factor of 3.3, the maximum being 3.4). The participation of the last four substitution sites becomes evident along these most relevant variables (see Table 3). Curiously, only site [Gln-1] seems to have here a marginal role and not [Pro-2], as stated by other authors. As expected, the L1O-CV returns a smaller AU-ROC value (0.929), but the other parameters (Accuracy, Sensitivity, ...) are maintained. When dealing with rules of order 2 the ROC curves become smoother as the range of cumulated molecular votes spreads.

The balanced leave-two-out (BL2O) procedure is a CV test suitable for dichotomized sets.[6,7] The name balanced comes from the fact that at every simulated cross-validation loop two molecules are left out simultaneously, one being of interest and the other being not. This prevents to generate all the combinations of molecular pairs but only $n_{int} \times n_{nint}$, the product of the number of items of interest (n_{int}) by the number of compounds labeled as not being of

Table 3. Most relevant rules of order 2 for the set of neuropeptides (*p*-values not greater than 10^{-10}). Only the relevant sites are listed. Dots stand for wildcard (*X*) rule elements. L stand for L-enantiomers and D stand for D-enantiomers. It becomes evident the participation of the last four sites, as stated by other authors. See text for more details

Rule	*p*-value	Vote	[Arg]	[Pro-2]	[Gln-2]	[Phe-1]	[Phe-2]	[Leu]
1	$1.9\cdot10^{-30}$	+	.	.	.	.	L	L
2	$2.4\cdot10^{-29}$	+	.	.	.	L	.	L
3	$3.4\cdot10^{-26}$	+	.	.	L	.	.	L
4	$2.9\cdot10^{-23}$	+	.	.	.	L	L	.
5	$1.6\cdot10^{-20}$	+	.	.	L	.	L	.
6	$1.9\cdot10^{-20}$	-	.	.	.	.	D	D
7	$8.3\cdot10^{-19}$	+	.	.	L	L	.	.
8	$1.1\cdot10^{-17}$	-	.	.	D	.	.	D
9	$1.7\cdot10^{-16}$	-	.	.	D	.	D	.
10	$1.7\cdot10^{-16}$	-	.	.	.	.	.	.
11	$2.1\cdot10^{-15}$	-	.	.	.	D	D	.
12	$2.3\cdot10^{-14}$	-	.	.	D	D	.	.
13	$4.0\cdot10^{-12}$	+	.	L	.	.	.	L
14	$8.0\cdot10^{-11}$	-	L	.	.	.	.	D
15	$8.0\cdot10^{-11}$	-	.	D	.	.	.	D

interest (n_{nint}). For every pair of molecules kept apart in a CV cycle, a couple of prediction votes is given (one for each left out item), and they are added up. At the end of the procedure, and for the sake of scaling purposes, the sum of votes for the relevant and non-relevant compounds has to be divided by n_{nint} and n_{int}, respectively. This gives a series of comparable votes that conforms a ranking. Additionally, during the BL2O cycles for each left out pair it is counted how many times a relevant compound had more votes than the other companion (correct internal classification case), the times both structures received the same number of votes (a tie), and the times the relevant item received less votes than the other (incorrect classification case). As it will be seen now, those counts are related to the AU-ROC value.[32,33]

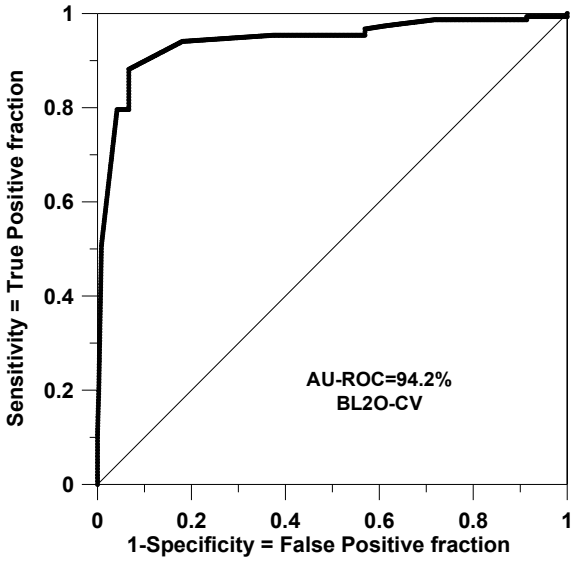

Figure 4. AU-ROC curve and AU-ROC area for the BL2O-CV experiment considering rules of order 2 and accepting only rules with *p*-value not greater than 10^{-10}. See text for details.

For the example explored here, the BL2O procedure required 54720 cycles (152×360). At each cycle normally 14 or 15 rules showed to be significant (*i. e.* *p*-value not greater than 10^{-10}). The AU-ROC for this ranking is 0.940 (Accuracy = 93.4 %, Sensitivity = 89.5 %, Specificity = 95.0 %, Precision = 88.3 %, Matthew's CC = 84.2 %, Hit rate at 5 % = 96.9 % again). Figure 4 shows the ROC curve for this CV experiment.

During the BL2O footage the molecular pairs were correctly sorted 50227 times, there were 2170 ties and in 2323 cases the left out pairs were misclassified. As said, from those figures an estimation of the AU-ROC value can be obtained: 50227 / 54720 = 91.8 % and adding half of the ties in the numerator this number increases up to 93.8 %.

Present results suggest that the obtained models are stable. This has been checked by means of two additional tests. First, a 5-fold CV was conducted, giving AU-ROC = 0.939, Accuracy = 92.6 %, Sensitivity = 88.8 %, Specificity = 94.2 %, Precision = 86.5 %, Matthew's CC = 82.4 %, Hit rate at 5 % = 100 % collecting 28 active compounds in the list of the first 28 ranked. This test has been repeated 10000 times, always obtaining quite similar results. In particular, the mean AU-ROC value for test compounds was 0.940 and ranged in the interval [0.925,0.949] for the 95% of the cases. Secondly, a series of 1000 randomization tests were done, each time scrambling the interest/non-interest labels at random and also selecting at random 256 compounds conforming the test set and the remaining ones acting as training elements. The models involved rules of order 2 and the threshold *p*-value was set now to 0.01 because no significant rules can be found if this threshold is set to the

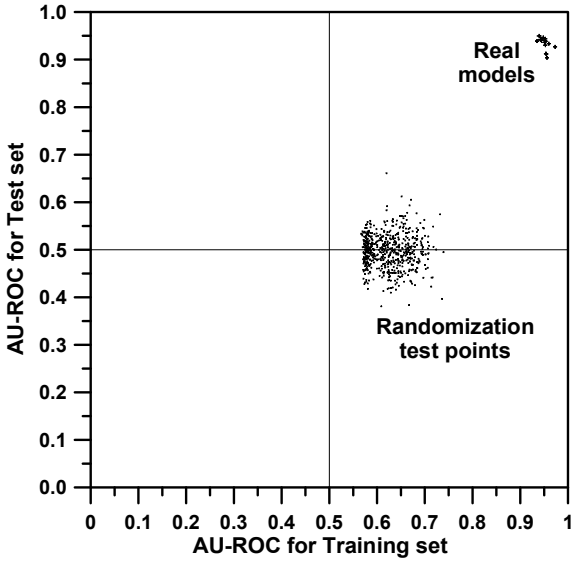

Figure 5. AU-ROC values for the 1000 randomization tests. The upper-right cloud of points corresponds to 100 correct calculations. This time the threshold *p*-value was set to 0.01. See text for details.

highly demanding 10^{-10} value. Working under the 1 % limit, in 377 cases (37.7 %) SSIR procedure was unable to get a model. The other models were able to find a maximum of 18 rules in one case. The range of achieved AU-ROC values for training in those cases was 0.567–0.739, whereas for the test set it was 0. 381–0.661. The best values were achieved by mere chance. Figure 5 shows the cloud composed by the 623 experiments with lead to select false rules and fake models (during training the method seeks for apparent profitable rules, but attached to low *p*-values that will not be useful for the external test set, as they are fake). Note in the graph how for the randomized test points all the AU-ROC values are surrounding the neutral value of 0.5 despite in training only values over 0.5 are obtained. In the same graphic the upper-right cloud of crosses was obtained when real (*i.e.* non-scrambled) random partitions of the set are trained and tested. This was done 100 times and the minimal AU-ROC values for training and test were 0.925 and 0.885, respectively.

Rules of order 3 or 4 gave similar results as above but did not increase the models performance. AU-ROC values for training and L1O-CV processes were 0.953 and 0.928 (rules of order 3) and 0.928 and 0.921 (rules of order 4). The number of total definable rules for each case were 672 and 2016, whereas the number of significant ones entering into the models were 41 and 37, respectively. No significant rules of order 5 were found (the *p*-value threshold was set to 10^{-10} in all the cases). The relevance of the rules of order 2 seems to corroborate the affirmation of Young and Hawkins: synergic interactions involving two molecular sites are the most relevant ones in this set. This feature can be attached to the pharmacophore concept: SSIR rules can help to elucidate the relevant sites and proper moieties they have to accommodate in a void scaffold.

CONCLUSIONS

It has been described SSIR, a systematic procedure useful to rank series of congeners and based on the simple superposition of rules which are obtained systematically. A multiple binary sites combinatorial example dataset has been explored in order to reveal the main features of SSIR. The results concerning the relevance of interactions of two molecular sites conform with previous literature conclusions, but using SSIR it has been shown that the finding is systematic. The Balanced Leave-two-out (BL2O) procedure has been also defined and applied in this context showing how the AU-ROC values can be estimated by the number of correct and incorrect pair classifications. This procedure, among other cross-validation tests, allowed concluding that the obtained ranking models are stable.

Acknowledgment. The author acknowledges the Generalitat de Catalunya (Departament d'Innovació, Universitats i Empresa) for the financial support given to the QTMEM (Química teòrica i Modelatge i Enginyeria Molecular) research group of the University of Girona (code 2014-SGR-1202). An anonymous referee is deeply acknowledged because he/she did an intensive analysis of the article, clearly helping to improve its contents and presentation.

REFERENCES

[1] J. Kolpak, P. J. Connolly, V. S. Lobanov, D. K. Agrafiotis, *J. Chem. Inf. Model.* **2009**, *49*, 2221.
[2] A. M. Wassermann, P. Haebel, N. Weskamp, J. Bajorath, *J. Chem. Inf. Model.* **2012**, *52*, 1769.
[3] D. K. Agrafiotis, J. J. M. Wiener, A. Skalkin, J. Kolpak, *J. Chem. Inf. Model.* **2011**, *51*, 1122.
[4] B. C. Duffy, L. Zhu, H. Decornez, D. B. Kitchen, *Bioorg. Med. Chem.* **2012**, *20*, 5324.
[5] J. L. Medina-Franco, B. S. Edwards, C. Pinilla, J. R. Appel, M. A. Giulianotti, R. G. Santos, A. B. Yongye, L. A. Sklar, R. A. Houghten, *J. Chem. Inf. Model.* **2013**, *53*, 1475.
[6] E. Besalú. *Int. J. Mol. Sci.* **2016**, *17*(6), 827.
[7] E. Besalú, L. Pogliani, J. V. de Julián-Ortiz in *Applied Chemistry and Chemical Engineering*, **2017**, **Vol. 4**. A. K. Haghi, L. Pogliani, E. A. Castro, D. Balköse, O. V. Mukbaniani, and C. H. Chia (Eds.). Apple Academic Press (AAP), Waretown, New Jersey.
[8] S. Monroc, E. Badosa, E. Besalú, M. Planas, E. Bardají, E. Montesinos, L. Feliu. *Peptides* **2006**, *27*(*11*), 2575.

[9] E. Badosa, R. Ferre, M. Planas, L. Feliu, E. Besalú, J. Cabrefiga, E. Bardají, E. Montesinos, *Peptides* **2007**, *28*(12), 2276.
[10] L. Feliu, G. Oliveras, A. D. Cirac, E. Besalú, C. Rosés, R. Colomer, E. Bardají, M. Planas, T. Puig, *Peptides*. **2010**, *31*(11), 2017.
[11] L. Eriksson, E. Johansson, N. Kettaneh-Wold, C. Wikström, S. Wold, *Design of Experiments. Principles and Applications*, **2000**, Umetrics Academy. Umea. Sweden.
[12] N. K. Terrett, *Combinatorial Chemistry*, 1998, Oxford University Press. Oxford.
[13] R. Carbó, E. Besalú, *J. Math. Chem.* **1993**, *13*, 331.
[14] E. Besalú, R. Carbó, *J. Math. Chem.* **1994**, *15*, 397.
[15] R. Carbó, E. Besalú, *Computers & Chemistry*, **1994**, *18*(2), 117.
[16] R. Carbó, E. Besalú in *Strategies and Applications in Quantum Chemistry: from Astrophysics to Molecular Engineering*. **Vol. 14**. Part 2. M. Defranceschi and Y. Ellinger (Eds.). Kluwer Ac. Pub., Amsterdam, 1996, pp. 229-248.
[17] E. Besalú, R. Ponec, J. V. de Julián-Ortiz, *Mol. Divers.* **2003**, *6*(2), 107.
[18] J. M. Barroso, E. Besalú. *Theochem*. **2005** *727*(1–3) 89.
[19] S. F. Yan, H. Asatryan, J. Li, Y. Zhou, *J. Chem. Inf. Model.* **2005**, *45*, 1784.
[20] A. Berkopec, *J. Discrete Algorithm*. **2007**, *5*, 341.
[21] E. Besalú, SSIR program v1.0, Girona, **2015**.
[22] A. Rácz, D. Bajusz, K. Héberger, *SAR QSAR Environ. Res.* **2015**, *26*, 683.
[23] M. Gütlein, C. Helma, A. Karwath, S. Kramer, *Mol. Inf.* **32** (2013) 516.
[24] D. Krstajic, L. J. Buturovic, D. E. Leahy, S. Thomas, *J. Cheminformatics* **2014**, *6*(10), 1.
[25] J. V. de Julián-Ortiz, E. Besalú, *Int. J. Mol. Sci.* **2006**, *7*(10) 456.
[26] E. Besalú, L. Vera, *J. Chil. Chem. Soc.* **2008**, *53*(3), 1576.
[27] J. V. de Julián-Ortiz, E. Besalú, R. García-Domenech, *Indian J. Chem. A*. **2003**, *42*(6) 1392.
[28] R. García-Domenech, J. V. de Julián-Ortiz, E. Besalú, *Mol. Divers.* **2006**, *10*(2), 159.
[29] J. X. Wang, A. M. Bray, A. J. DiPasquale, N. J. Maeji, H. M. Geysen, *Int. J. Peptide Prot. Res*. **1993**, *42*, 384.
[30] J. X. Wang, A. J. DiPasquale, A. M. Bray, N. J. Maeji, H. M. Geysen, *Bioorg. Med. Chem. Lett*. **1993**, *3*(3), 451.
[31] S. S. Young, D. M. Hawkins, *J. Med. Chem.* **1995**, *38*(14), 2784.
[32] E. Besalú, J. V: de Julián-Ortiz, and L. Pogliani in *Quantum Frontiers of Atoms and Molecules*. M. V. Putz (Ed.). NOVA Publishing Inc. New York, 2010, pp 589−605.
[33] S. J. Mason and N. E. Graham, *Q. J. R. Meteorol. Soc.* **2002**, *128*, 2145.